



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/975,945	10/11/2001	Charles Paclat	THEOR-205.1-US	9612
24972 7590 10/16/2007 FULBRIGHT & JAWORSKI, LLP 666 FIFTH AVE NEW YORK, NY 10103-3198			EXAMINER KHATRI, ANIL	
			ART UNIT 2191	PAPER NUMBER
			MAIL DATE 10/16/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/975,945	PACLAT, CHARLES	
	Examiner	Art Unit	
	Anil Khatri	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 15 August 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-18 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-18 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

Art Unit: 2191

DETAILED ACTION

The response filed on 8/15/07 under 37 CFR 1.131 is sufficient to overcome the USPN 7,086,065 reference and rejection for claim 1-18 under 35 USC 102(e) have been withdrawn by the examiner.

Applicant's arguments/amendment necessitated the new ground(s) of rejection presented in this Office action.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-16 are rejected under 35 U.S.C. 102(e) as being anticipated by *Goodwin et al* USPN 6,199,195

Regarding claim 1

Goodwin et al teaches

Art Unit: 2191

analyzing a business domain to determine functional requirements of business domain (column 11, lines 36-55, the meta object facility defines and manipulates a set of meta models. An example of a model is a set of business objects that makes up a running software application. An example of a meta model is the language used to describe the object classes. More specifically, each model adaptor 310 loaded into the repository adaptor tool 312 imports a domain specific meta model (i.e., a logical model) and translates it into a unified model. The goal of the schema server 316 is to describe data structures and behaviors of various systems using the unified modeling language, so that the client application 338 can make high level decisions based on the data descriptions (unified models). Because information returned from the schema server 316 describes data, the returned values can also be referred to as meta data (and sometimes M2, distinguishing it from the schema repository 314, whose meta data can be referred to as M1). The meta data is distributed as, for example, CORBA interface objects as defined in the object management group's (OMG) Unified Modeling Language (UML) 1.1 specification documents, incorporated herein by Reference);

transforming functional requirements into an EJB component model (columns 7-8, lines 53-67 and 1-5, The code generator 210 allows application developers to define the system definition 208, which describes how new services are integrated into a business object framework. The system definition 208 provides application developers with a flexible approach for generating code from a data model. When a data model is loaded into the code generator 210, the model is converted first by the model adaptor 204 into the unified

Art Unit: 2191

models 206. The unified models are then passed into each template (i.e., each component of the system definition 208) for code generation); and

building an EJB component in accordance with EJB component model that encompass the business functionality of business domain (column 12, lines 25-35, the repository adaptor tool 312, of course, uses logical models from various modeling tools to generate the unified models. Just as each pen stroke on a blueprint represents a unique structural feature of building, each object within a unified model represents a unique structural feature of a software system. More specifically, the objects describe all aspects of systems that have been modeled. One or more objects represent each component of a system, such as a structural feature or a behavioral feature. A single object easily defines a majority of system components, while some components must be represented with a graph of objects).

Regarding claim 2

Goodwin et al teaches

modifying functional requirements by a user; and repeating the steps (b) and (c) to provide a parallel development process (column 7, lines 8-21, during development-time, the disclosed system also allows object developers to define how these objects are mapped to enterprise resources such as to object oriented databases, world wide web resources, and relational databases. The disclosed system provides object oriented access to distribute heterogeneous enterprise resources. The mapping of objects can initially be performed using any of a number of commercially available modeling tools, the output of which are the logical models 202. These logical models 202 are

Art Unit: 2191

composed in a variety of modeling languages specific to the modeling tools with which they are created. At development-time, these logical models 202 are all converted to the unified modeling language, regardless of the modeling language in which they are composed, making up the unified models 206).

Regarding claims 3 and 11

Goodwin et al teaches

EJB components are extensible and configurable (column 8, lines 20-30, one advantage of the present embodiment is providing methods for developing the extensible application frameworks that can rapidly integrate object services as they become available from commercial sources or are developed as domain specific features by the developer. This level of extensibility and tailoring is unparalleled in heretofore available systems. The code generator 210 and the ability to rapidly integrate new object services (proxy, persistent state, domain specific services, etc.) into business objects provides a capability that is unique to the present embodiment and its system definition 208, and to integrating object models with object behaviors within a business object framework).

Regarding claims 4 and 5

Goodwin et al teaches

functional requirements include data and process model of business domain (column 12, lines 42-47, While each building block in the schema repository 314 is an object, the

Art Unit: 2191

composition of the schema repository 314 reflects specific goals of the schema server 316. The goal of the unified model is to describe high level application business objects that are familiar to developers. The meta data are objects that describe the application business objects in a domain-independent matter).

Regarding claims 6 and 12

Goodwin et al teaches

step of analyzing includes the step of generating a list of inputs, each input identifying a resource that relate to business domain (column 12, lines 36-55, The meta object facility defines and manipulates a set of meta models. An example of a model is a set of business objects that makes up a running software application. An example of a meta model is the language used to describe the object classes. More specifically, each model adaptor 310 loaded into the repository adaptor tool 312 imports a domain specific meta model (i.e., a logical model) and translates it into a unified model. The goal of the schema server 316 is to describe data structures and behaviors of various systems using the unified modeling language, so that the client application 338 can make high level decisions based on the data descriptions (unified models). Because information returned from the schema server 316 describes data, the returned values can also be referred to as meta data (and sometimes M2, distinguishing it from the schema repository 314, whose meta data can be referred to as M1). The meta data is distributed as, for example, CORBA interface objects as defined in the object management group's (OMG) Unified Modeling Language (UML) 1.1 specification documents, incorporated herein by Reference).

Regarding claim 7

Goodwin et al teaches

the step of generating eFunction matrix from list of inputs (column 12, lines 49-57, The unified model also represents structure of external data sources such as relational databases. In order to stay true to object oriented philosophy, the philosophy of meta information, and the correct representation of physical data, an object oriented system and a relational table system are represented in the unified model. Software classes maintained by developers, such as interfaces defined in Interface Definition Language (IDL) files or classes, or C++ header files are the definitions of applications of business objects).

Regarding claim 8

Goodwin et al teaches

step of transforming transforms functional requirements using an unified modeling language (UML) tool to generate EJB component model (column 8, lines 42-60, Referring next to FIG. 3, shown is a detailed block diagram showing particular features and aspects of the software system described above in further detail. Shown are a number of modeling tools 302, 304, 306 both data modeling 302 and object modeling 304, 306, defining data within a database 308 or defining objects and relating these objects to the data within the database 308. These definitions are referred to herein as logical models. Also shown are a plurality of model adapters 310 for defining a translation of the logical models of the modeling tools 302, 304, 406 into unified

Art Unit: 2191

models, expressed in a unified modeling language, such as Unified Modeling Language (UML). A repository adaptor tool 312 takes the logical models generated by the modeling tools 302, 304, 306 (logical models 202 (FIG. 2)) and the model adaptors 310 as inputs and generates the unified models 206 (FIG. 2) in the unified modeling language. The unified models 206 (FIG. 2) are stored in a schema repository 314 accessed by a schema server 316. The unified models 206 (FIG. 2) from the repository adaptor tool 312 are received by the schema server 316 and stored within the schema repository 314).

Regarding claim 9

Goodwin et al teaches

EJB component model includes a plurality of EJB classes(column 8, lines 6-19 developer defines which unified models are passed to each template or sub-template. This can be either a class, package, subclass, or any other unified model-defined object. Other unified models are accessed using control structures. The syntax of the templates supports any number of control structures followed by a block code. Control structures and code can be nested within one another. The present system also offers the unique advantage of user-defined language extensions. This tailor ability allows developers to implement their own template syntax (parser and interpreter) in which templates can be implemented. The preferred syntax for the templates is JavaScript, although any of a number of well known or custom syntaxes can be used with relatively equal effectively.

Art Unit: 2191

Goodwin et al teaches

Claim 10

step of building builds EJB component from at least one of the following class stereotypes: Belonging, Session, Entity, Configurable Entity, Business Policy and Workflow (column 4, lines 1-5, An "object class" is a set of data (attributes) and functional capabilities (routines) encapsulated into a single logical entity. For example, a "drug treatment" class may be characterized by a "dosage" attribute and an "administer-medication" routine.

Regarding Claims 13 and 16

Goodwin et al teaches

building includes the step of generating relational mappings and deployment descriptors (column 7, lines 8-21, During development-time, the disclosed system also allows object developers to define how these objects are mapped to enterprise resources such as to object oriented databases, world wide web resources, and relational databases. The disclosed system provides object oriented access to distribute heterogeneous enterprise resources. The mapping of objects can initially be performed using any of a number of commercially available modeling tools, the output of which are the logical models 202. These logical models 202 are composed in a variety of modeling languages specific to the modeling tools with which they are created. At development-time, these logical models 202 are all converted to the unified modeling language, regardless of the modeling language in which they are composed, making up the unified models 206).

Art Unit: 2191

Regarding Claims 14 and 15

Goodwin et al teaches

generating end-user documentation (column 4, lines 19-25, "a state full session bean..."); (column 8, lines 63-65 and column 9, lines 1-3, Also shown are a number of operating frameworks 318, such as Java RMI, OMG, and COM (e.g., DCOM) standard database services. Each of these operating frameworks defines a plurality of services 320, such as the event services, naming services, proxy services and the like. Also shown is a custom service 322, such as might be developed for a particular application and/or for a particular end user by a developer);

developing unit tests to test EJB component; and (column 8, lines 20-30, one advantage of the present embodiment is providing methods for developing the extensible application frameworks that can rapidly integrate object services as they become available from commercial sources or are developed as domain specific features by the developer. This level of extensibility and tailoring is unparalleled in heretofore available systems. The code generator 210 and the ability to rapidly integrate new object services (proxy, persistent state, domain specific services, etc.) into business objects provides a capability that is unique to the present embodiment and its system definition 208, and to integrating object models with object behaviors within a business object framework); and

generating a reference implementation of said EJ-B component (column 6, lines 22-28, Together, these components comprise a system and method for designing and authoring source code objects, through which services can be composed within a complex business

Art Unit: 2191

object framework, and for generating business objects with all implementations and behaviors within a composed object service framework.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 17-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Goodwin et al* USPN 6,199,195 in view of *Lee et al* USPN 6,944,680.

Regarding claims 17-18

Goodwin et al teaches EJB component but does not teach explicitly smart component having at least one of following Smart feature: SmartKey, SmartHandle and SmartValue. However, Lee et al teaches (column 6 , lines 1-32 , Turning now to FIG. 2, there is illustrated a SmartHandle of the present invention. A SmartHandle 200 is a Java class, and it is comparable 210 and serializable 220. The SmartHandle 200 has the following attributes: KeyClassName (of type String), HomeClassName (of type String) and HomeName (of type String). In addition, the SmartHandle contains, by reference, a reference to an instance of a SmartKey 230, a reference to a transient instance of an EJBObject 240 and a reference to a transient instance of a Handle 250. Transient attributes are nullified when the SmartHandle 200 is persisted. The SmartHandle 200 has, in addition to its inherited methods and lifecycle methods, a getEJBObject()

Art Unit: 2191

method that returns the EJBObject and an equals() method used for comparison purposes. The SmartHandle 200 maintains an Entity EJB object relationship by using a combination of the proxy pattern, EJB Handle, and primary key. A SmartHandle stores the name of the EJB Home class, or factory, from which an Entity EJB was generated and from which it can be re-instantiated. It also maintains an instance of the SmartKey 230 that describes the primary key for the database column(s) to which the entity object is mapped. A single method invocation instantiates the object that is referred to by the SmartHandle 200 by first looking up the Home class using the Java Naming and Directory Interface (JNDI), using reflection to obtain the ejbFindByPrimaryKey method, and then invoking that method with the persisted SmartKey 230. Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention was made to incorporate EJB component in eBusiness with Smart features. The modification would have been obvious because one of ordinary skill in the art would have been motivated to combine teaching EJB component for different Smart features and values to achieve efficiency with the component.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Anil Khatri whose telephone number is 571-272-3725. The examiner can normally be reached on M-F 8:30-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2191

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.


ANIL KHATRI
PRIMARY EXAMINER